cat event\_history.csv | grep -i "serdar" | grep -i "terminate" | grep -Eo "i-.{17}" | sort | uniq | tee result.txt

#!/bin/bash

#

# This script creates a new user on the local system.

# You will be prompted to enter the username (login), the person name, and a password.

# The username, password, and host for the account will be displayed.

# Make sure the script is being executed with superuser privileges.

if [ "$UID" -ne 0 ]

then

echo "Please run as root"

exit 1

fi

# Get the username (login).

read -p "Please write your name: " USER\_NAME

# Get the real name (contents for the description field).

read -p "Enter a comment: " COMMENT

# Get the password.

read -sp "Enter the password" PASSWORD

# Create the account.

useradd -c "$COMMENT" -m $USER\_NAME

# Check to see if the useradd command succeeded.

# We don't want to tell the user that an account was created when it hasn't been.

if [[ $? -ne 0 ]]

then

echo "The account could not be created"

exit 1

fi

# Set the password.

echo $PASSWORD | passwd --stdin $USER\_NAME

# Check to see if the passwd command succeeded.

if [[ $? -ne 0 ]]

then

echo "The password couldn't be set"

exit 1

fi

# Force password change on first login.

passwd -e $USER\_NAME

# Display the username, password, and the host where the user was created.

echo "$USER\_NAME"

echo "$PASSWORD"

sed -i "s/ec2-private\_ip/$(grep "PrivateIpAddress" info.json | head -1 | cut -d'"' -f 4)/g" terraform.tf

echo -e $(cat certificate.pem) > new.pem

cat auth.log | grep "Failed password for invalid user" | awk '{print $11}' | sort | uniq -c > result.txt

mkdir data data1

[10:40](https://aws12-tr.slack.com/archives/C03NYKEU43V/p1663098017197429)

touch file{1..9}

[10:41](https://aws12-tr.slack.com/archives/C03NYKEU43V/p1663098070413609)

mkdir myfolder

[10:41](https://aws12-tr.slack.com/archives/C03NYKEU43V/p1663098074400589)

cd myfolder

[10:41](https://aws12-tr.slack.com/archives/C03NYKEU43V/p1663098105668239)

touch file{11..19

#!/bin/bash

# Check if we are root privilage or not

if [ "$UID" -ne 0 ]

then

echo "Please run as root"

exit 3

fi

# Which files are we going to back up. Please make sure to exist /home/ec2-user/data file

backup\_files="/home/ec2-user/data /home/ec2-user/myfolder"

# Where do we backup to. Please crete this file before execute this script

dest="/mnt/backup"

# Create archive filename based on time

time=$(date +"%Y\_%m\_%d\_%H\_%M")

hostname=$(hostname -s)

archive\_file="$hostname-$time.tgz"

# Print start status message.

echo "archive process is started"

# Backup the files using tar.

tar -czvf $dest/$archive\_file $backup\_files

# Print end status message.

echo "backup is finished"

# Long listing of files in $dest to check file sizes.

ls -l /mnt/backup